

Chapter 4

Bootstrap 3

Bootstrap:

Bootstrap is a really useful CSS and JavaScript library. It is a single CSS file and a single JavaScript file. Inside the CSS file there is a bunch of stuff that we get for free.

Components are the bigger pieces.

We can either download it to our machine and work/load locally (just use link tag for this after moving the 'bootstrap.css' file to the work area) or we can include it online by using,

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BVYiISIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/
K68vbdEjh4u" crossorigin="anonymous">
```

To include Bootstrap 4,

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
integrity="sha384-
9aIt2nRpC12Uk9gS9baD1411NQApFmC26EwAOH8WgZl5MYXxHfc+NcPb1dKGj7Sk"
crossorigin="anonymous">
```

- We can find any info and any choice of any element on the getbootstrap.com,

Bootstrap 3,

<https://getbootstrap.com/docs/3.3/css/>

<https://getbootstrap.com/docs/3.3/components/>

Bootstrap 4,

<https://getbootstrap.com/docs/4.5/getting-started/introduction/>

<https://getbootstrap.com/docs/4.5/components/>

- To simply overwrite the styles, we can create a 'style' section in the head of our HTML file,

```
<style type="text/css">
  .btn-success {
    background-color: blue;
  }
</style>
```

- Jumbotron

Jumbotron is the big header like thing with an explanation and a button. It takes the 100% of the contain that it is inside of,

<https://getbootstrap.com/docs/3.3/components/#jumbotron>

<https://getbootstrap.com/docs/4.5/components/jumbotron/>

- Grid System

To handle the kinda problematic thing above in other words to put some things to the center of the webpage or make it just 1/3 of the page etc. we normally use 'Grid System'. But for a basic solution we can also use the following 'container' class to handle it:

- Container

A container is a predefined class in Bootstrap that used with 'div' tags. It automatically creates a container with predefined useful margines and width etc.

- Forms

The advancedly modified version of the form we have seen back in ChapterOne:

<https://getbootstrap.com/docs/3.3/css/#forms>

```
.form-group | <div class="form-group">
```

A label-input pair is better used within "form-group" class to group'em together and add some spacing from the rest.

`.form-control | <input type="email" class="form-control" id="InputEmail" placeholder="email">`

Specifies the style of the input box and its effect when selected.

`.form-inline | <form class="form-inline">`

The class that used with the 'form' tag in order to make the form inline.

`.checkbox | <div class="checkbox">`

When we put label-input(checkbox) pair together inside this div class, the label there turns into a clickable text which is awesome

- We can place 'input' into 'label',

```
<label>
  <input type="checkbox"> Check me out
</label>
```

- Shift + Tab in Sublime Text takes the selected lines 1 indent backwards.

- Nav Bars

<https://getbootstrap.com/docs/3.3/components/#navbar>

<https://getbootstrap.com/docs/4.5/components/navbar/>

In the 'Default Navbar' at first we will realize that the hamburger menu (shows up when we shrink the window) and the 'Dropdown' menus are not working. To make them work we need to include 'JavaScript' bootstrap file, (The following is Bootstrap 3)

```
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNlcpD7Txa"
crossorigin="anonymous"></script>
```

And then also include a 'jquery' script over, by copying the most recent/appropriate version and pasting it like we included the Bootstrap JS file with <script> tags,

<https://code.jquery.com/>

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"
integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0="
crossorigin="anonymous"></script>
```

- Tag & classes for creating the navbar,

```
<nav class="navbar navbar-default"></nav>
```

- It's better if we put the things in the navbar into a container div!

-* We put the menu buttons inside of the following div in order to push'em into hamburger menu when the window is shrunk,

```
<div class="collapse navbar-collapse" id="my-menus">
  <ul class="nav navbar-nav">
    <li><a href="#">About</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
  <ul class="nav navbar-nav navbar-right">
    <li><a href="#">Sign Up</a></li>
    <li><a href="#">Login</a></li>
  </ul>
</div>
```

-** The hamburger button,

```
<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#my-menus" aria-expanded="false">
  <span class="sr-only">Toggle navigation</span>
  <span class="icon-bar"></span>
  <span class="icon-bar"></span>
  <span class="icon-bar"></span>
</button>
```

- In the Hamburger button**, 'data-target' ATR must match the 'id' ATR in the collapse div*!

- The header section of the navbar,

```
<div class="navbar-header"></div>
```

Brand and Hamburger menu button are placed inside of this header part.

- The left side menu tags of the navbar,

```
<ul class="nav navbar-nav"></ul>
```

- The right side menu tags of the navbar,

```
<ul class="nav navbar-nav navbar-right"></ul>
```

The Grid System

Maybe it's the main reason that people prefer using bootstrap! As I shrink the window, apart from the content being scaled, it adapts into mobile or tablet mode easily and visually looking awesome,

<https://getbootstrap.com/docs/3.3/css/#grid>

<https://getbootstrap.com/docs/4.5/layout/grid/>

- It scales up to 12 columns. Anytime we use the grid, we must include it into a container! It consists of row and column divs. And so an example usage of grid system is as following,

```
<div class="container">
  <div class="row">
    <div class="col-lg-6 blue">COL LG 6</div>
    <div class="col-lg-6 blue">COL LG 6</div>
  </div>
</div>
```

- There are 4 sizes to be used in the Grid System:

1. xs: For mobile size
2. sm: For tablet size
3. md: For small screens or windows (laptop,PC)
4. lg: For big screens and windows

We can specify portions for each of these sizes.

- If we don't specify the lg size explicitly, it will take it from the md element,

```
<div class="col-md-3 col-sm-6 blue">TOUR DATE!</div>
<div class="col-md-3 col-sm-6 blue">TOUR DATE!</div>
<div class="col-md-3 col-sm-6 blue">TOUR DATE!</div>
<div class="col-md-3 col-sm-6 blue">TOUR DATE!</div>
```

So here when it's lg size, 4 of them will be on the same line as in md; when it's sm size they will get separated 2 by 2; and finally when it's xs they will go 1 by 1 because we didn't specify it and when we don't specify a smaller size or any size, it will take full 12 columns which means 1 element at 1 line.

- Btw above the number coming after the size specifies how many columns the element will be on,

```
<div class="col-lg-6 blue">COL LG 6</div>
<div class="col-lg-6 blue">COL LG 6</div>
```

Here it will be on 6 columns, which means there will be 2 of'em on the line when the size is lg

- We can make nested grids.

- `<nav class="navbar navbar-inverse"></nav>`

Gives us a reversed navbar (with a deep background and light text by default)

- `<div class="thumbnail"></div>`

Creates a standart box around the element inside of it which is generally an image. If we don't use it around an image even in a grid, image won't fit properly and will go as its size!

- Ctrl + D: Selects all the instances of a selected text one by one on each push within Sublime Text 3, which will implicitly duplicate the cursor. It's better to choose the text from beginning to end by letting the cursor be at the end not at on the front!

- If we want to get rid of the changing size images -especially hight; because when we apply thumbnail width is fixed but hight is not- then we can see the code here,

<https://codepen.io/nax3t/pen/MJwpdb>

- To solve the white space bug along the image places,

This bug can be fixed by adding a class to the div with class of row, name it flex, then make a css rule like so:

```
.flex {
  display: flex;
  flex-wrap: wrap;
}
```

- Glyphicons,

<https://getbootstrap.com/docs/3.3/components/>

No Glyphicons in Bootstrap 4,

<https://www.glyphicons.com/>

<https://primer.style/octicons/>

Also a nice alternative, for both 3 and 4,

<https://fontawesome.com/>

- Glyphicons has the feature of text in terms of colors; so color ATR will change its color directly and so the font-size will change its size.

- To make the navbar fixed on the top while scrolling down, just add class,

`navbar-fixed-top`

After making this we will realize that our content stayed behind it which means on the top we need padding, so we will give our body 70px top padding which is the hight of the navbar by default,

`padding-top: 70px;`

- It is important that we include our own style file after the line we include bootstrap, otherwise we cannot overwrite them!

- When we try to overwrite styles of elements and are not able to do, it is always the best solution to go the page and checking the most specific ATR that effects the element we want to change by 'inspecting element'

- In Navbars

Div class 'container' stretches the bar all along the screen but keeps the content in the container,

```
<nav class="navbar navbar-default">
  <div class="container">
    </div>
</nav>
```

But div class 'container-fluid' keeps both the content and the bar in the container, which is the default actually,

```
<nav class="navbar navbar-default">
  <div class="container-fluid">
    </div>
</nav>
```

- background-image: url();
background-size: cover;
background-position:center;

- In order to make the background image cover all the file,

```
html {  
    height: 100%;  
}
```

- An awesome shadow that contributes depth to the texts on the page,

```
text-shadow: 0px 4px 3px rgba(0,0,0,0.4),  
            0px 8px 13px rgba(0,0,0,0.1),  
            0px 18px 23px rgba(0,0,0,0.1);  
/*text-shadow: offset-x offset-y radius color;*/  
https://developer.mozilla.org/en-US/docs/Web/CSS/text-shadow
```

- If we want our bootstrap styled website to be responsive on mobile then we should to add the following meta tag to our <head> element, above the <title> tag:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

- To fix the appearance of the <hr> on mobile devices, add max-width: 90%; to the hr selector,

```
hr {  
    width: 400px;  
    max-width: 99%;  
}
```